Literature Review of Machine Learning Methods

Jonathan Gillett

July 2015

Abstract

Machine Learning methods have numerous applications in scientific fields and in highly commercialized industries. The ability to organize data into sensible groupings that can be more easily understood and analyzed is fundamental to understanding the vast amount of data available. The focus of this literature review is on Machine Learning methods for data clustering and dimensionality reduction and how these methods are improved using intelligent methods.

1 Introduction

Machine Learning is a branch of Computer Science that has since evolved from the study of pattern recognition and computational theory into a highly diverse and widely applied area of research. This evolution has begun to empower society to take advantage of the almost limitless amount of data that has become available[12, 15].

Machine Learning focuses on the creation of algorithms, often rooted in artificial intelligence methods; in order to learn from, model, and make predictions based on data.[3]. These algorithms frequently involve the analysis and understanding of large data sets, often referred to as "Big Data", which are seemingly insurmountable using traditional statistical methods[12]. For example, in 2007 there was approximately 281 exabytes of data consumed on the Internet, and by 2011 this amount had increased tenfold[8]. Most of this data is stored digitally in electronic media, thus providing huge potential for the development of automatic data analysis, classification, and retrieval techniques.

The increase in the sheer volume and the variety of data requires advances in methodology to automatically understand, process, and summarize the data. As such, this has created a huge demand for Machine Learning techniques, which are applied to the data for performing clustering analysis, dimensionality reduction, and modelling to better understand and predict trends. Clustering analysis involves algorithms for grouping or clustering objects according to measured or perceived intrinsic characteristics or similarity; making it possible to find patterns which would otherwise be indistinguishable[8]. Furthermore, dimensionality reduction is also an essential step in Machine Learning, and is used to reduce the negative affects of the "curse of dimensionality", which often arises

when an algorithm does not scale well to high-dimensional data, requiring time or memory that increases exponentially based on the dimensions of the data[4].

Given the importance of Machine Learning techniques to the scientific community it is the purpose of this review paper to provide an overview of clustering and dimensionality reduction algorithms used in Machine Learning, with a focus on the applications of intelligent methods to improve their accuracy and robustness.

2 Review of Clustering Methods

Clustering, also known as cluster analysis, is essential to discovering the natural groupings of a set of patterns, points, or objects within the data. All clustering methods can be described as follows: Given a representation of N objects, find K groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are low[8].

As a result of the similarity criteria for clustering, it can be expected that clustering is a subjective entity that often is in the eye of the beholder and whose significance and interpretation requires domain knowledge. As a result of the requirement for domain knowledge, it has resulted in thousands of clustering algorithms being published for use, and that continue to appear[8]. The rapid increase in clustering algorithms has created a need for an underlying robust clustering method that uses intelligent methods to encapsulate some of the domain knowledge of the expert without requiring a new algorithm. The vast amount of literature citing the use of clustering algorithms speaks to the importance of clustering in data analysis.

The intent of this section is to provide a review of K-means clustering methods, discuss the development and refinements of clustering algorithms, and lastly to point out some of the emerging and useful research directions that utilize intelligent methods to improve the accuracy and robustness of clustering.

2.1 *Some Methods for Classification and Analysis of Multivariate Observations & Least Squares Quantization in PCM*

K-means has a rich and diverse history as it was independently discovered in multiple scientific fields prior to it's formalization and independent publication by Lloyd (published in 1982)[11] and MacQueen (published in 1967)[13]. Despite the fact that K-means was first proposed over 50 years ago, it is still one of the most widely used algorithms for clustering[8].

In K-means the data is clustered by first determining a set of centroid points. The algorithm then finds a partition, such that the squared error between the empirical mean of a cluster and the centroid points in the cluster is minimized[13, 11].

As outlined independently by MacQueen and Lloyd in their publications of the K-means clustering algorithm, it can be formally described in mathematical terms as follows.

Let $X = x_i, i = 1, \ldots, n$ be the set of n d-dimensional points to be clustered into a set of K clusters, $C = c_k, k = 1, \ldots, K$. K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_k be the mean of the cluster c_k, the squared error between μ_k and c_k is defined as follows.

$$J(c_k) = \sum_{x_i \in c_k} \|x_i - \mu_x\|^2 \tag{1}$$

Where the optimization criteria of K-means is to minimize the sum of the squared error over all K clusters.

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_x\|^2 \tag{2}$$

2.1.1 Reflections

The main reasons for the continued popularity of K-means is due to the ease of implementation, simplicity, and efficiency[8]. The subsequent 50 years of development in clustering algorithms have been based on the equations 1 and 2 for K-means.

While K-means has been the most popular clustering algorithm, the original algorithm does not take into consideration any information about the problem domain; limiting it's accuracy and robustness in comparison to subsequent improvements[8].

Lastly, it is important to note that the K-means optimization criteria for the final clustering, as described in equation 2, which minimizes the sum of the squared error over all K clusters is **NP-HARD**[5].

2.2 *The Uniqueness of a Good Optimum for K-Means*

In this paper the author, Meila, focused on addressing the most critical issue in K-means; that the optimization criteria for the final clustering is **NP-HARD**. The paper provides proofs that it is possible to find a "good" clustering C of a data set that is not far from the optimal, by measuring the "goodness" of the distortion of K-means clustering.

It is significant as the paper proves the spectral bounds on the distance $d(C, C^{opt})$, where C is a cluster and C^{opt} the optimal, that it is possible to find a "good" clustering of C without having to perform the **NP-HARD** computation required to find the best clustering[14]. Thus, by using a measure of "goodness" a unique and compact cluster of near-optimal clusterings can be found without needing to perform the computationally expensive **NP-HARD** operation[14].

2.2.1 Reflections

This paper is significant in that it provides the first proven method of reducing the daunting **NP-HARD** complexity of the K-means clustering method; which is often amplified given the sheer magnitude of the data being analyzed in most Big Data applications of Machine Learning techniques.

By providing a proven method for finding a suitable clustering that is a "good" fit, without requiring evaluating all possible clusters, the results of this paper made the continued widespread adoption of K-means even more applicable given the increased performance that it can have when applied to increasingly large data sets.

2.3 *Estimating the Number of Clusters in a Data Set via the Gap Statistic*

The K-means algorithm requires three user-specified parameters: number of clusters K, cluster centroids initialization, and a distance metric[17]. These are all dependent on the domain knowledge of the expert, with the most critical choice parameter being K, the number of clusters.

While there is no perfect mathematical criterion that exists, there are a number of heuristic approaches available for choosing K. Typically, K-means is run independently for different values of K and the partition that appears the most meaningful to the domain expert is then selected. This poses a problem as the different initializations can often lead to vastly different final clustering because K-means only converges to local minimal[17].

In order to mitigate this time-consuming and computationally expensive process, Tibshirani, et al. introduced a method of estimating the number of clusters in a group, referred to as the "gap statistic", to limit the requirement of domain knowledge[17]. As a result, this enhancement greatly increased the robustness of K-means, in addition to enhancing the intelligence of the "unsupervised" learning of K-means, requiring less domain knowledge to create the optimal results.

2.3.1 Reflections

The method of the "gap statistic" introduced by Tibshirani, et al. made it possible to perform K-means without requiring domain knowledge about the number of clusters in the data. However, the remaining parameters essential to K-means: the cluster centroids initializations, and distance metrics still require domain knowledge.

The use of additional intelligent methods such as evolutionary computation, Genetic Algorithm (GA), would make it possible to make K-means purely unsupervised, thus relying on the optimization abilities of GA, rather than requiring a domain expert to specify parameters. The GA could be used to enhance the "gap statistic" by optimizing the remaining parameters of the k-means clustering.

2.4 *K-means Clustering of Proportional Data using L1 Distance*

The third parameter required for K-means is the distance metric, most frequently the Euclidean distance is used as the metric for computing the distance between points and cluster centroids[9]. Thus, as a result, K-means gravitates towards finding spherical or ball-shaped clusters in data, however this is not always the desired outcome, especially for higher-dimensional data.

To address this issue, Kashima et al. introduced the application a new distance metric for K-means known as $L1$, or more commonly referred to as *Manhattan distance*. The $L1$ distance, is simply the sum of the absolute differences of their Cartesian coordinates projected onto the determined coordinate with fixed step size[9]. More formally, the $L1$ distance can be described as follows where p, q are vectors on a fixed Cartesian coordinate system.

$$d_1(p,q) = \|p-q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{3}$$

Where (p, q) are vectors defined as follows.

$$p = (p_1, p_2, \ldots, p_n) \quad \textbf{and} \quad q = (q_1, q_2, \ldots, q_n) \tag{4}$$

2.4.1 Reflections

The research done by Kashima et al. improves the third parameter of K-means, the distance comparison metric, by using the $L1$ distance method instead of the default Euclidean distance, this is yet another improvement to K-means in addition to the application of the "gap statistic" introduced by Tibshirani, et al.[17].

However, both the "gap statistic" and the $L1$ distance metric introduced by Tibshirani, et al. and Kashima et al.[17, 9] lack the application of soft methods for determining the distances and thus membership of data within clusters.

There is much potential for future applications of clustering methods that use fuzzy logic and set theory for the distance metrics and clustering memberships, to categorize clusters rather than relying solely on a crisp membership within clusters based on the distance.

2.5 *Fast Accurate Fuzzy Clustering through Data Reduction*

One of the major improvements to the original K-means method published independently by Lloyd and MacQueen[11, 13] is the fuzzy C-means method of clustering. As it's name implies, fuzzy C-means applies the intelligent soft computing methods of fuzzy logic and set theory to provide fuzzy membership within clusters rather than the traditional crisp sets that are used to denote membership within a cluster.

Fuzzy C-means was first proposed by Dunn in 1973[6] and later improved by Bezdek in 1984[2], and extended the robustness of K-means by making it possible for each data point to be a member of multiple clusters with a membership value for each (soft assignment). Despite the benefits of the soft assignments of fuzzy C-means, it's application was limited due to the performance overhead until recently when Eschrich et al. published their method on performing fast fuzzy C-means through data reduction[7].

In their paper Eschrich et al. provide a method of improving the performance of fuzzy C-means with their algorithm, brFCM, which performs a reduction and aggregation of similar examples and then uses a weighted exemplar in the clustering process[7]. The brFCM algorithm performs quantization of the data, reducing it from a continuous data type to one that has an acceptable level of quantization at the expense of a loss in precision. Following the quantization, the data is aggregated to combine identical feature vectors into a single weighted exemplar. The quantization then performs fine-grained binning of the data making it easier to find identical feature vectors for the aggregation process[7].

2.5.1 Reflections

While the performance of the method demonstrated by Eschrich et al. provided a 59 to 290 improvement over a traditional implementation of fuzzy C-means using brFCM[7], the performance is still not optimal for clustering of Big Data due to the overhead of the fuzzy set operations[8].

However, the use of brFCM for fuzzy C-means demonstrably provides more optimal clustering than the default operations and the intelligent methods of fuzzy set theory make it possible to have accurate clusters without having to rely on "gap statistics" to find the optimal number of clusters[17].

3 Review of Dimensionality Reduction Methods

Dimensionality reduction is essential in Machine Learning to reduce the negative affects of the "curse of dimensionality", which often arises when an algorithm does not scale well to high-dimensional data, requiring time or memory that increases exponentially based on the dimensions of the data[4]. Dimensionality reduction is used to reduce the number of dimensions of the data, making it more easily comprehensible as a reduced number of components which encapsulate the majority of the variance in the data.

Dimensionality is a quintessential operations in Machine Learning, with numerous applications in other fields. The most widely used methods in Machine Learning is Principal Component Analysis (PCA)[8, 4, 3], but more recently groundbreaking research has been done by leading researchers in Deep Learning to create a more intelligent algorithm for dimensionality reduction known as t-distributed Stochastic Neighbor Embedding (t-SNE).

The intent of this section is to provide a literature review of the most widely used dimensionality reduction methods, PCA and t-SNE, and to discuss the

latest research in this area.

3.1 *Principal Component Analysis*

Dimensionality reduction is an essential operation in Machine Learning, PCA provides a method of reducing the dimensionality of higher dimensional data into it's "principal components". The essential concept of PCA is a method to transforms a number of possibly correlated variables into a smaller number of variables, which are called principal components[16, 1].

The principal components make it possible to perform a visual (in 1 - 3 dimensions) and even quantitative examination (for higher dimensions) of the reduced dimensional data set. This allows the domain expert to spot trends, patterns and outliers in the data, far more easily than would have been possible without performing the principal component analysis[16].

In computing the principal components the first task of PCA is to identify a new set of orthogonal coordinate axes through the data, this is achieved by finding the direction of maximal variance through the coordinates in the N dimensional space[16]. These coordinates are then used to project the N-dimensional data onto a K-dimensional linear subspace that minimizes the *reconstruction error*, which is the sum of the squared L_2-distances between the original and projected data. The resultant projection of this operation captures the most variance in the data within the orthogonal principal components of the linear subspace[16, 1, 15]. Following the selecting of the initial principal component such that it meets this criteria, the PCA method then proceeds to obtain further principal coordinate (axis) which are both orthogonal to the other previously selected principal components, and are the next best direction maximizing the variance in the data, chosen from directions which are orthogonal to the first principal component[16, 15].

3.1.1 Reflections

PCA is often referred to as one of the most important results to come out of the field of applied linear algebra[16], and for good reason. PCA has proven to be very versatile, and is one of the oldest and most popular techniques used in dimensionality reduction for multivariate analysis[1]. With PCA it is possible to reduce a seemingly incomprehensible multivariate problem into one that can have the majority of the variance explained by only a few principal components using only common linear algebraic operations[16].

Despite the longevity and popularity of PCA, it does not incorporate any intelligent methods, such as evolutionary computation, or fuzzy logic in order to improve it's robustness and provide soft selection of principal components. However, as will be discussed in the proceeding sections, *t-SNE* has proven to be a strong contender to PCA with it's inception rooted in Deep Learning.

3.2 *A General Framework for Increasing the Robustness of PCA-based Correlation Clustering Algorithms*

Expanding upon the long history and popular adoption of PCA, Kriegel, et al. focused on making PCA more robust to outliers in the data. By it's nature PCA is rather sensitive to outliers, if a small fraction of these points do not correspond to the correct correlation of the cluster, the algorithms are usually misled or even fail to detect the correct results[10].

Kriegel, et al. evaluated the influence of outliers on PCA and proposed a general framework for increasing the robustness of PCA, and found in many cases when performing PCA that if the correct subspace of the corresponding cluster contains noise the subspace determination process will be misled[10].

In order to address the influence of outliers and noise on PCA Kriegel, et al. proposed a weighting function which significantly reduces the negative impact of outliers on PCA. First, a weighting function is applied to the points when computing the covariance matrix, this is done in order to weight points that are potential outliers, or noisy, lower than points that are potential cluster members. After the weighting is applied a method for selecting a suitable number of neighbors for each cluster member or cluster is applied by micro-adjusting the parameters to avoid sudden drops in the explained variance and choosing significantly different parameters for points in the data set[10].

3.2.1 Reflections

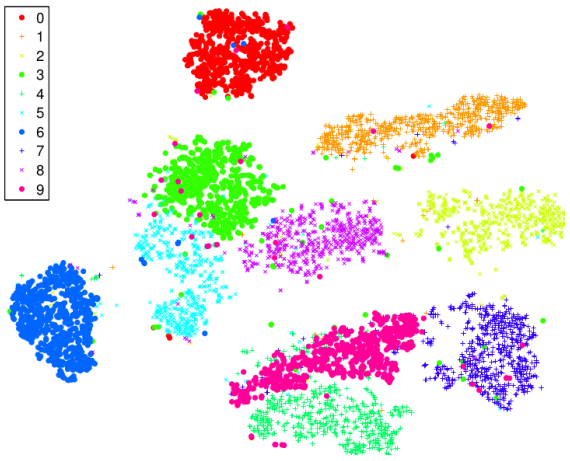
Almost all correlation clustering algorithms suffer from an arbitrary selection of points in the local neighborhood of cluster members, and are heavily dependent on the initial choice made. The process of selecting the correct cluster members from which the subspace of a cluster is determined by applying PCA often has a significant influence on the amount of outliers in the data.

Kriegel, et al. focused on not solving the problem of making more suitable selections of cluster members, but instead focused on trying to lessen the negative influence of outliers and noisy data on PCA. The weighting and micro-adjusting method provides a useful, but unfortunately non-intelligent method of improving the robustness of PCA. It is interesting to note what potential benefits their method would have if it was enhanced by fuzzy logic and set theory to make the selection of cluster members soft rather than crisp.

3.3 *Visualizing Data Using t-SNE*

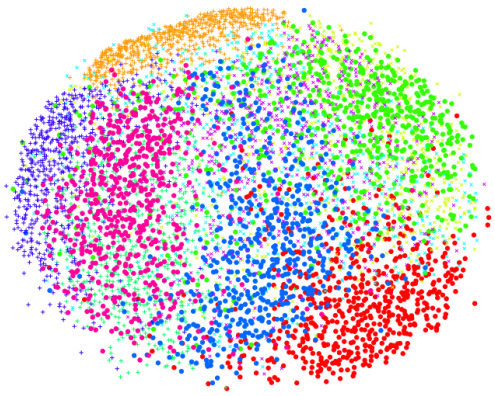
One of the most significant advances in dimensionality reduction since the inception of PCA has been t-distributed Stochastic Neighbor Embedding (t-SNE) created by leading Machine Learning researchers, Van der Maaten and Hinton. t-SNE makes it possible to visualize high-dimensional data giving each data point a location in a two or three-dimensional map, making it much easier to optimize and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map[18].

The origin of t-SNE is one born out of dissatisfaction, after investigating all of the leading methods of non-linear dimensionality reduction (e.g. PCA) that aim to preserve the local structure, Van der Maaten and Hinton were dissatisfied with the weak performance on real high-dimensional data[18]. Motivated to come up with a revolutionary new method of dimensionality reduction, t-SNE was created and tested against many other leading dimensionality reduction algorithms using real high-dimensional data. On almost all of the data sets tested by Van der Maaten and Hinton the visualizations produced by t-SNE were significantly better, providing a much clearer identification of clusters.



.

Figure 1: Clustering of 6,000 handwritten digits from MNIST data set using t-SNE[18]



.

Figure 2: Clustering of 6,000 handwritten digits from MNIST data set using linear dimensionality reduction (e.g. PCA)[18]

The failings of linear techniques such as PCA for high-dimensional data, is that they focus on keeping the low-dimensional representations of dissimilar data points far apart. This is problematic as for very high dimensional that lies on or near a low-dimensional, non-linear manifold it is usually more important to keep the low-dimensional representations of very similar data points close together[18]. This is something that is not typically possible with a linear mapping such as used by PCA.

Stochastic Neighbor Embedding (SNE), the basis of t-SNE, is based on the concept of converting high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities. The following equation describes the similarity of data point x_j to data point x_i as the conditional probability, $p_{j|i}$, that x_i would pick x_j as its neighbor if the neighbours were picked proportionally based on their probability density under a Gaussian distribution centered at x_i[18].

Mathematically, this conditional probability $P_{j|i}$ used by SNE is given as follows, where σ_i is the variance of the Gaussian that is centered on the data point x_i.

$$P_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{5}$$

SNE, performs this calculation to determine the pairwise data point similarities, which are the two conditional probabilities $P_{j|i}$ and $Q_{j|i}$. Using the conditional probability calculations for the pairwise similarities, SNE aims to find a low-dimensional data representation that minimizes the mismatch between $P_{j|i}$ and $Q_{j|i}$.

Using this method SNE constructs reasonably good visualizations, however it is hampered by a cost function that is difficult to optimize and by a problem referred to by Van der Maaten and Hinton to as the "crowding problem"[18]. t-SNE alleviates these problems by using a unique cost function that differs from SNE in two ways. First the t-SNE cost function uses a symmetrized version of the SNE cost function with simpler gradient calculations. The second and most significant improvement is that t-SNE uses a Student-t distribution, rather than the Gaussian distribution shown in equation 5, to compute the similarity between two points in the low-dimensional space.

3.3.1 Reflections

Despite the drastic improvement in the clustering results and accuracy for t-SNE it has severely limiting computational and memory complexities that are quadratic in the number of data points analyzed[18]. While t-SNE performed exceedingly well in comparison to existing methods this limiting computational and memory complexity makes it infeasible to apply the standard version of t-SNE to data sets that contain many more than approximately 10,000 points[18].

Given this limitation it is unlikely that t-SNE will find greater acceptance with real world data, as Machine Learning methods are being applied to increasingly larger and larger data sets every year[12]. Perhaps there is still the

potential for t-SNE to find greater application if further improvements can be made to address it's computational and memory complexities, but for now it is likely that the existing methods of K-means and PCA will continue to be used.

4 Conclusion

The numerous applications for Machine Learning are obvious, the importance of creating robust and accurate algorithms for performing clustering and dimensionality reduction are critical to understanding and analyzing the ever-increasing amount of information at our disposal. In this review paper we compiled a thorough list of the fundamental techniques and algorithms for the most widely used methods of clustering and dimensionality reduction, K-means and PCA.

After laying the foundation for the two most common methods, we then provided a thorough review of each of the latest improvements that have been made to these algorithms. In the first section for clustering algorithms, the paper *The Uniqueness of a Good Optimum for K-Means*, by Meila, provided a value performance enhancement to K-means making it possible to find a "good" clustering that is not far from the optimal, reducing the daunting NP-HARD complexity of K-means for finding the optimal[14]. In *Estimating the Number of Clusters in a Data Set via the Gap Statistic*, Tibshirani provided a method to enhance the intelligence required to execute K-means, estimating the number of clusters using the "gap statistic", to limit the requirement of domain knowledge for the number of clusters[17]. Lastly, the most significant paper, *Fast Accurate Fuzzy Clustering through Data Reduction*, by Eschrich et al. provided an innovative method for improving the performance of fuzzy C-means with their algorithm, brFCM. Where, fuzzy C-means applies the intelligent soft computing methods of fuzzy logic and set theory to provide fuzzy membership within clusters rather than the traditional crisp sets[7].

Lastly, following a literature review of clustering methods we provided a thorough review of dimensionality reduction, giving an explanation of the most commonly used method, Principal Component Analysis (PCA). While PCA is one of the oldest and most popular techniques and has proven to be very versatile for dimensionality reduction, there have been no major advances in dimensionality reduction until very recently. Dissatisfied with the poor results of existing methods on real data, leading Machine Learning researchers, Van der Maaten and Hinton, created the revolutionary t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm. t-SNE provided a drastic improvement in the clustering results and accuracy for but unfortunately is severely limited due to the rapid increase in the computational and memory complexity as the number of data points analyzed increase, making it infeasible for very large data sets[18].

In conclusion, it is significant to note the longevity of the K-means clustering and PCA dimensionality reduction algorithms, after nearly 50 years, these algorithms are still widely utilized for solving real-world problems and continue to be improved by leading Machine Learning experts. We feel that their is still

much potential to improve these methods further or to create new algorithms, as demonstrated by Van der Maaten and Hinton with t-SNE, that utilize intelligent methods to further improve the robustness and accuracy of these critical Machine Learning algorithms.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.

[3] Thomas G Dietterich. Machine learning for sequential data: A review. In *Structural, syntactic, and statistical pattern recognition*, pages 15–30. Springer, 2002.

[4] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[5] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V Vinay. Clustering large graphs via the singular value decomposition. *Machine learning*, 56(1-3):9–33, 2004.

[6] JC DUNN. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[7] Steven Eschrich, Jingwei Ke, Lawrence O Hall, and Dmitry B Goldgof. Fast accurate fuzzy clustering through data reduction. *Fuzzy Systems, IEEE Transactions on*, 11(2):262–270, 2003.

[8] Anil K Jain. Data clustering : 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[9] Hisashi Kashima, Jianying Hu, Bonnie Ray, and Moninder Singh. K-means clustering of proportional data using l1 distance. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[10] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. A general framework for increasing the robustness of pca-based correlation clustering algorithms. In *Scientific and Statistical Database Management*, pages 418–435. Springer, 2008.

[11] Stuart P Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[12] Steve Lohr. The age of big data. *New York Times*, 11, 2012.

[13] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[14] Marina Meilă. The uniqueness of a good optimum for k-means. In *Proceedings of the 23rd international conference on Machine learning*, pages 625–632. ACM, 2006.

[15] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[16] Mark Richardson. Principal component analysis. *Mathematical Modelling and Scientific Computing, University of Oxford, Oxford, UK*, 2009.

[17] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[18] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.